UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/797,238	03/10/2004	Rajeev B. Rajan	MSFT-2924/306986.01	2995

41505 7590 03/08/2007

WOODCOCK WASHBURN LLP (MICROSOFT CORPORATION)
CIRA CENTRE, 12TH FLOOR
2929 ARCH STREET
PHILADELPHIA, PA 19104-2891

EXAMINER	
VAUTROT, DENNIS L	
ART UNIT	**PAPER NUMBER**
2167	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	03/08/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	Application No.	Applicant(s)	
	10/797,238	RAJAN ET AL.	
	Examiner	Art Unit	
	Dennis L. Vautrot	2167	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *15 December 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-35* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-35* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *10 March 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Specification

1. The amendments to the specification dated 15 September 2006 have been received and are accepted.

Response to Amendment

2. The applicants' amendment, filed 15 December 2006, has been received, entered into the record and considered.

3. As a result of the amendment, claims 1, 14, 15, 19, 20, 22, and 24 are amended. Claims 1 – 35 are pending in the application.

Response to Arguments

4. Applicant's arguments with respect to claims 1 – 35 have been considered but are moot in view of the new ground(s) of rejection.

5. Regarding the arguments for claims 14 and 19, after further consideration of **Narang**, it appears that the check for read and write locks occurs after the interception of certain file system calls, see page 2, paragraph [0020]. Examiner interprets this to mean that the determination of whether read or write access available is in response to receiving the file system statement. The rejection provides additional explanation.

6. Regarding the arguments for claims 1 and 24, after further consideration of **Narang**, it appears that acquiring either a read or write lock occurs after the interception of certain file system calls, see page 2, paragraph [0020]. Examiner interprets this to mean that the acquisition is in response to receiving the file system statement. The rejection provides additional explanation.

Claim Rejections - 35 USC § 102

7. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

8. Claims 14, 17 – 19, and 22 - 23 are rejected under 35 U.S.C. 102(e) as being anticipated by **Narang** (hereinafter **Narang**, US 2005/0091287).

9. Regarding claim 14, **Narang** discloses a method for locking and isolation of a file system statement including a call to open an item, a call to read from the item, and a call to close the item (See page 2, paragraph [0019] "The application can now access the external data file directly using standard file-system API calls. Typical file-system API calls are 'file-open' 'file read' and 'file delete'." And see page 5, paragraph [0076]

"This covers the case where an updated has modified and then closed the file after releasing any file locks..."), the method comprising:

receiving the file system statement (See page 2, paragraph [0020] "The DLFF 10 intercepts certain of the file system API calls 70 issued by the application. The file-system API calls that are intercepted include file-open, file-rename, and file-delete calls.");

in response to receiving the file system statement, determining if a read lock is available for a row of a data table corresponding to the item (See page 2, paragraph [0020] "If the external data file 20 is referenced in the database 80, the DLFF 10 is responsible for enforcing referential-integrity constraints and access-control requirements defined for the data file 20." This referential-integrity and access-control enforcement is being broadly interpreted to include determining if a read lock is available on the row of a data, as part of its integrity and control functions. Also, it appears that the file system statement has been received, as it follows the interception of the file system API call, therefore this determination is interpreted to take place in response to the file system statement.);

if not, then failing the open (See page 2, paragraph [0021] "If the validation fails, the file-open request is rejected."); and

if so, then acquiring the read lock on the row (See page 2, paragraph [0021] "Once the access has been authorized by a valid token, the application interacts directly with the file system for delivery of the external file without the need for the DLFF to further control file access." And see page 3, paragraph [0044] "In a typical update,

access to the file is requested and the file is locked." In other words, the access would be locked once it was determined read access was available.)

10. Regarding claim 17, **Narang** additionally discloses acquiring the read lock on a filestream field [picture field] of the row. (See page 2, paragraph [0021] "Once the access has been authorized by a valid token, the application interacts directly with the file system for delivery of the external file without the need for the DLFF to further control file access." And see page 3, paragraph [0044] "In a typical update, access to the file is requested and the file is locked." In other words, the access would be locked once it was determined read access was available. And see page 2, paragraph [0016] "The name column 83 typically contains a string, the department column an integer, while the picture column would contain a reference to an image stored in one of a number of external data files 20." The picture field in the reference is the same thing as the filestream field of the claim.)

11. Regarding claim 18, **Narang** additionally discloses a computer readable medium having computer-executable instructions for performing the steps recited in claim 14. (See page 10, paragraph [0164] "In particular, the software may be stored in a computer readable medium, including the storage devices described below.")

12. Regarding claim 19, **Narang** teaches a method for locking and isolation of a file system statement including a call to open an item (See page 2, paragraph [0019]

"Typical file-system API class are 'file-open'..."), a call to write to the item (See page 6, paragraph [0088] "In step 260 of FIG. 3A, the application 210 requests a write access token from the DB2 agent 220 via a SQL SELECT statement."), and a call to close the item (See page 5, paragraph [0076] "This covers the case where an updater has modified and then closed the file after releasing any file locks..."), the method comprising:

receiving the file system statement (See page 2, paragraph [0020] "The DLFF 10 intercepts certain of the file system API calls 70 issued by the application. The file-system API calls that are intercepted include file-open, file-rename, and file-delete calls.");

in response to receiving the file system statement, determining if a write lock is available for a row of a data table corresponding to the item (See page 2, paragraph [0014] "On the other hand, WRITE PERMISSION BLOCKED provides data recovery for a file with a reference in a DATALINK column. However, a user cannot update the file while the file is currently linked.");

if not, then failing the open (See page 3, paragraph [0042] "However, access to the file may be denied."); and

if so, then acquiring a write lock on the row. (See page 3, paragraph [0044] "In a typical update access to the file is requested and the file is locked.")

13. Regarding claim 22, **Narang** additionally discloses acquiring the write lock on a filestream field [picture field] of the row. (And see page 3, paragraph [0044] "In a typical

update, access to the file is requested and the file is locked." And see page 2, paragraph [0016] "The name column 83 typically contains a string, the department column an integer, while the picture column would contain a reference to an image stored in one of a number of external data files 20." The picture field in the reference is the same thing as the filestream field of the claim.)

14. Regarding claim 23, **Narang** additionally discloses a computer readable medium having computer-executable instructions for performing the steps recited in claim 19. (See page 10, paragraph [0164] "In particular, the software may be stored in a computer readable medium, including the storage devices described below.")

Claim Rejections - 35 USC § 103

15. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

16. Claims 1, 12, and 13 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Nojima** (US 2004/0199540) in view of **Narang et al.** (hereinafter **Narang**, US 2003/0069902).

17. Regarding claim 1, **Nojima** teaches a method for executing a file system statement in the context of a transaction (See page 3, paragraph [0041] "The DBMS executes the process in a unit of transaction."), the file system statement including a call to open an item [DB managed contents], one of a call to read from the item [DB managed contents], and a call to write to the item [DB managed contents], and a call to close the item [DB managed contents] (See page 5, paragraph [0075] "A data read function is for acquiring data of a DB managed contents 243." And see page 6, paragraph [0082] "A data write function is for performing additional write or update of data for the DB managed contents 243."), the method comprising:

associating the file system statement with the transaction (See page 5, paragraph [0070] "A transaction settling function is to settle link registration, link delete and content update for an external contents.").

Nojima does not explicitly disclose receiving the file system statement and in response to receiving the file system statement, starting the transaction by acquiring either a read lock or a write lock on a data table row corresponding to the item.

However, **Narang** teaches receiving the file system statement (See page 2, paragraph [0020] "The DLFF 10 intercepts certain of the file system API calls 70 issued by the application. The file-system API calls that are intercepted include file-open, file-rename, and file-delete calls."); and

in response to receiving the file system statement, starting the transaction by acquiring either a read lock or a write lock on a data table row corresponding to the item. (See page 1, paragraph [0004] "...a transaction coordinator typically ensures a

consistent view by locking out readers of meta-data as well as file data until the transaction is committed." This is referring to locking out database access during the transaction, which occurs by readers of the meta-data, which would have been initiated after intercepting the file system statement mentioned above.)

It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Nojima** with those of **Narang** because both address the field of database consistency related to external objects, and by including the read and write locks as disclosed in **Narang**, consistency control can be maintained while transactions are being processed. It is for this reason that one of ordinary skill in the art would have been motivated to include receiving the file system statement and in response to receiving the file system statement, starting the transaction by acquiring either a read lock or a write lock on a data table row corresponding to the item.

18. Regarding claim 12, **Nojima** teaches a method substantially as claimed.

Nojima does not explicitly disclose starting the transaction by acquiring one of a read lock and a write lock on a filestream field of the row.

However, **Narang** teaches starting the transaction by acquiring one of a read lock and a write lock [cannot update the file] on a filestream field [picture column] of the row. (See page 2, paragraph [0014] "On the other hand, WRITE PERMISSION BLOCKED provides data recovery for a file with a reference in a DATALINK column. However, a user cannot update the file while the file is currently linked." And see page 2, paragraph [0016] "The name column 83 typically contains a string, the department

column an integer, while the picture column would contain a reference to an image stored in one of a number of external data files 20." The picture field in the reference is the same thing as the filestream field of the claim.)

It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Nojima** with those of **Narang** because both address the field of database consistency related to external objects, and by including the read and write locks as disclosed in **Narang**, consistency control can be maintained while transactions are being processed. It is for this reason that one of ordinary skill in the art would have been motivated to include starting the transaction by acquiring one of a read lock and a write lock on a filestream field of the row.

19. Regarding claim 13, **Nojima** additionally discloses a computer readable medium having computer-executable instructions for performing the steps recited in claim 1. (See page 2, paragraph [0038] "The user client 10 includes a CPU 11, a memory 12, a network interface 13 and a secondary storage 14 which are coupled with each other through a bus 19....Stored on the secondary storage are 14 an operating system (hereinafter called an OS) 141, a user program 142, a DBMS external file linking library (hereinafter called an FLL) 143 and a registration target contents 144 to be registered from the user program into a database.")

20. Claim 2 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Nojima** in view of **Narang** as applied to claim 1 above, and further in view of **Reed et al.**

(hereinafter **Reed**, US 7,035,874). **Nojima** and **Narang** teach starting the transaction by acquiring one of a read lock and a write lock on a data table row. (See **Narang** page 1, paragraph [0004] "...a transaction coordinator typically ensures a consistent view by locking out readers of meta-data as well as file data until the transaction is committed." This is referring to locking out database access during the transaction.)

Nojima and **Narang** do not explicitly disclose the row includes a user defined type corresponding to the item.

However, **Reed** teaches the row includes a user defined type corresponding to the item. (See column 3, lines 48-51 "SQL also supports user defined types (UDT) and user defined methods (UDM). In one implementation, DBMS 100 supports SQL and includes support for UDT's and UDM's." and column 4, lines 41 – 43 "The cursor locks a selected portion of a table in DBMS 100 including the indicated media object (e.g., locking one or more rows).")

It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Nojima** and **Narang** with those of **Reed** because they all address the field of database consistency related to external objects, and by including the UDT's as disclosed in **Reed**, a more robust method is created because a variety of types of complex data can be referred to. It is for this reason that one of ordinary skill in the art would have been motivated to include the row includes a user defined type corresponding to the item.

21. Claims 3 – 5 and 7 – 11 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Nojima** in view of **Narang** as applied to claim 1 above, and further in view of **Bamford et al.** (hereinafter **Bamford**, US 5,870,758).

22. Regarding claim 3, **Nojima** and **Narang** teach a method substantially as claimed. **Nojima** and **Narang** fail to teach associating a second statement [one or more] with the transaction.

However, **Bamford** teaches associating a second statement with the transaction (See column 1, lines 20-21 "A logical unit of work that is comprised of one or more database language statements is referred to as a transaction." This shows that multiple statements can be associated with the transaction.)

It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Nojima** and **Narang** with those of **Bamford** because they all comprise database methods for transaction concurrency and by including more than one statement within the transactions, it becomes more realistic as database transactions usually comprise more than one statement, as is well known in the art. It is for this reason that one of ordinary skill in the art would have been motivated to include associating a second statement with the transaction.

23. Regarding claim 4, **Narang** additionally discloses the second statement being another file system statement. (See page 2, paragraph [0019] "The application 30 can

now access the external data file 20 directly using standard file-system API calls 70. Typical file-system API calls 70 are 'file-open', 'file-read' and 'file-delete'.")

24. Regarding claim 5, **Nojima** additionally discloses the second statement being a transactional query language statement. (See page 4, paragraph [0062] "Described in (5) is a SELECT statement for acquisition of an access handle for accessing an external contents." Select is an example of a transactional query language statement.)

25. Regarding claim 7, **Nojima** and **Narang** disclose a method substantially as claimed.

Nojima and **Narang** do not explicitly disclose acquiring the read lock on the row comprises acquiring a read committed view of the row.

However, **Bamford** discloses acquiring the read lock on the row comprises acquiring a read committed view of the row. (See column 5, line 66 – column 6, line 7 "According to an embodiment of the invention, the SQL92 isolation levels READ COMMITTED and SERIALIZABLE are implemented using what shall be referred to hereinafter as 'Read Consistent Mode'. Read Consistent Mode is characterized by two rules. First, every statement executed by transaction sees only (1) changes that were committed to the database by a particular set of committed transactions (the 'snapshot set') and (2) changes made by the transaction itself.")

It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Nojima** and **Narang** with those of **Bamford**

because they all comprise database methods for transaction concurrency and by including the read committed view as disclosed in Bamford, the database has an additional tool needed to retain consistency. It is for this reason that one of ordinary skill in the art would have been motivated to include acquiring the read lock on the row comprises acquiring a read committed view of the row.

26. Regarding claim 8, **Nojima** and **Narang** disclose a method substantially as claimed.

Nojima and **Narang** do not explicitly disclose acquiring the write lock on the row comprises acquiring a write lock that will prevent another transaction from accessing the row while the transaction is being processed. However, **Bamford** discloses acquiring the write lock on the row comprises acquiring a write lock that will prevent another transaction from accessing the row while the transaction is being processed. (See column 6, lines 9 – 12 "Second, update transactions lock the rows they write, and hold those locks until the transactions complete. A lock on any given row may be held by only one transaction at a time.") It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Nojima** and **Narang** with those of **Bamford** because they all comprise database methods for transaction concurrency and by including the write lock on the row as disclosed in Bamford, the database has an additional tool needed to retain data consistency. It is for this reason that one of ordinary skill in the art would have been motivated to include acquiring the

write lock on the row comprises acquiring a write lock that will prevent another transaction from accessing the row while the transaction is being processed.

27. Regarding claim 9, **Nojima** and **Narang** disclose a method substantially as claimed.

Nojima and **Narang** do not explicitly disclose acquiring the write lock on the row comprises acquiring a write lock that will prevent a non-transacted file system statement from accessing the row while the transaction is being processed.

However, **Bamford** discloses acquiring the write lock on the row comprises acquiring a write lock that will prevent a non-transacted file system statement from accessing the row while the transaction is being processed. (See Column 3 lines 12 – 17 "Dirty write: only one transaction can hold a write lock on a specific data item. To preclude dirty writes, write locks must be held until a transaction commits. Virtually all database systems always prevent dirty writes, in order to support transaction rollback and prevent totally unpredictable and chaotic results.")

It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Nojima** and **Narang** with those of **Bamford** because they all comprise database methods for transaction concurrency and by including non-transacted file system statement restriction from row access as disclosed in **Bamford**, the database has an additional tool needed to retain data consistency. It is for this reason that one of ordinary skill in the art would have been motivated to include acquiring the write lock on the row comprises acquiring a write lock that will prevent a

non-transacted file system statement from accessing the row while the transaction is being processed.

28. Regarding claim 10, **Nojima** and **Narang** disclose a method substantially as claimed.

Nojima and **Narang** do not explicitly disclose acquiring the write lock on the row comprises acquiring a write lock that will prevent another statement within the transaction from writing to the row.

However, **Bamford** discloses acquiring the write lock on the row comprises acquiring a write lock that will prevent another statement within the transaction from writing to the row. (See column 6 lines 48 – 52 "For example, Read Consistent Mode with transaction-level snapshots may be implemented such that the snapshot set for all queries in a transaction includes all of the transactions that committed prior to the beginning time of the transaction." Because this is based on the transaction level, between individual queries, other statements will not be able to be write until the entire transaction has finished.)

It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Nojima** and **Narang** with those of **Bamford** because they all comprise database methods for transaction concurrency and by including a write lock as disclosed in Bamford, the database has an additional tool needed to retain data consistency. It is for this reason that one of ordinary skill in the art would have been motivated to include acquiring the write lock on the row comprises

acquiring a write lock that will prevent another statement within the transaction from writing to the row.

11. The method of claim 11, **Nojima** and **Narang** disclose a method substantially as claimed.

Nojima and **Narang** do not explicitly disclose acquiring the write lock on the row comprises acquiring a write lock that will enable another statement within the transaction to read from the row.

However **Bamford** discloses acquiring the write lock on the row comprises acquiring a write lock that will enable another statement within the transaction to read from the row. (See column 6, line 38-41 "For example, Read Consistent Mode with query-level snapshots may be implemented such that the snapshot set for each query includes all transactions that committed prior to the beginning time of the query." Because this is based on individual queries, rather than transactions, queries that occurred within a transactions will not be locked from being read.) It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Nojima** and **Narang** with those of **Bamford** because they all comprise database methods for transaction concurrency and by including a write lock as disclosed in Bamford, the database has an additional tool needed to retain data consistency. It is for this reason that one of ordinary skill in the art would have been motivated to include acquiring the write lock on the row comprises acquiring a write lock that will enable another statement within the transaction to read from the row.

29. Claim 6 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Nojima** in view of **Narang** as applied to claim 1 above, and further in view of **Ponnekanti** (6,606,626).

Nojima and **Narang** teach a method substantially as claimed.

Nojima and **Narang** do not explicitly disclose determining whether starting the transaction will result in a conflict; if so, then resolving the conflict according to a conflict resolution scheme; and if not, then starting the transaction.

However, **Ponnekanti** teaches determining whether starting the transaction will result in a conflict (See column 4, lines 3 – 7 "The instant duration lock is a mechanism that allows the client requesting the lock to see whether there exists a conflicting lock already held on the row (i.e. from another concurrent transaction.); if so, then resolving the conflict according to a conflict resolution scheme (See column 4, lines 10-18 where a conflict resolution scheme is described); and if not, then starting the transaction [delete]. (See column 4, lines 7- "If no conflict is found, the 'lock instant' request will be granted and the client will know that the 'delete' has committed.")

It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Nojima** and **Narang** with that of **Ponnekanti** because they all comprise database methods for transaction concurrency and by including a conflict resolution scheme as described in **Ponnekanti**, the method becomes more robust and able to deal with errors. It is for this reason that one of ordinary skill in the art would have been motivated to include determining whether

starting the transaction will result in a conflict; if so, then resolving the conflict according to a conflict resolution scheme; and if not, then starting the transaction.

30. Claim 15 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Narang** as applied to claim 14 above, and further in view of **Reed**.

Narang discloses determining if the read lock is available for a row of a data table corresponding to the item (See page 2, paragraph [0021] "...the DLFF validates the authorization token to determine whether or not to pass the file-open request through to the native file system.")

Narang fails to disclose the table includes a user-defined type.

However, **Reed** discloses the table includes a user-defined type. (See column 3, lines 48-51 "SQL also supports user defined types (UDT) and user defined methods (UDM). In one implementation, DBMS 100 supports SQL and includes support for UDT's and UDM's." and column 4, lines 41 – 43 "The cursor locks a selected portion of a table in DBMS 100 including the indicated media object (e.g., locking one or more rows).")

It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Narang** with those of **Reed** because they both address the field of database consistency related to external objects, and by including the UDT's as disclosed in **Reed**, a more robust method is created because a variety of types of complex data can be referred to. It is for this reason that one of ordinary skill in

the art would have been motivated to include a user defined type corresponding to the item.

31. Claim 16 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Narang** as applied to claim 14 above, and further in view of **Bamford.**

Narang fails to disclose acquiring the read lock on the row comprises acquiring a read committed view of the row.

However, **Bamford** discloses acquiring the read lock on the row comprises acquiring a read committed view of the row. (See column 5, line 66 – column 6, line 7 "According to an embodiment of the invention, the SQL92 isolation levels READ COMMITTED and SERIALIZABLE are implemented using what shall be referred to hereinafter as 'Read Consistent Mode'. Read Consistent Mode is characterized by two rules. First, every statement executed by transaction sees only (1) changes that were committed to the database by a particular set of committed transactions (the 'snapshot set') and (2) changes made by the transaction itself.")

It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Narang** with those of **Bamford** because they both comprise database methods for transaction concurrency and by including the read committed view as disclosed in **Bamford**, the database has an additional tool needed to retain consistency. It is for this reason that one of ordinary skill in the art would have been motivated to include acquiring the read lock on the row comprises acquiring a read committed view of the row.

32. Claim 20 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Narang** as applied to claim 19 above, and further in view of **Reed**.

Narang discloses determining if the write lock is available for a row of a data table corresponding to the item (See page 2, paragraph [0014] "On the other hand, WRITE PERMISSION BLOCKED provides data recovery for a file with a reference in a DATALINK column. However, a user cannot update the file while the file is currently linked.")

Narang does not explicitly disclose the table includes a user-defined type.

However, **Reed** discloses the table includes a user-defined type. (See column 3, lines 48-51 "SQL also supports user defined types (UDT) and user defined methods (UDM). In one implementation, DBMS 100 supports SQL and includes support for UDT's and UDM's." and column 4, lines 41 – 43 "The cursor locks a selected portion of a table in DBMS 100 including the indicated media object (e.g., locking one or more rows).")

It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Narang** with those of **Reed** because they both address the field of database consistency related to external objects, and by including the UDT's as disclosed in **Reed**, a more robust method is created because a variety of types of complex data can be referred to. It is for this reason that one of ordinary skill in the art would have been motivated to include a user defined type corresponding to the item.

33. Claim 21 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Narang** as applied to claim 19 above, and further in view of **Bamford** (US 5,870,758).

Narang discloses a method substantially as claimed.

Narang fails to disclose acquiring the write lock on the row comprises acquiring a write lock that will prevent another statement from accessing the row while the statement is being processed.

However, **Bamford** discloses acquiring the write lock on the row comprises acquiring a write lock that will prevent another statement from accessing the row while the statement is being processed. (See Column 3 lines 12 – 17 "Dirty write: only one transaction can hold a write lock on a specific data item. To preclude dirty writes, write locks must be held until a transaction commits. Virtually all database systems always prevent dirty writes, in order to support transaction rollback and prevent totally unpredictable and chaotic results.")

It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Narang** with those of **Bamford** because they both comprise database methods for transaction concurrency and by including non-transacted file system statement restriction from row access as disclosed in **Bamford**, the database has an additional tool needed to retain data consistency. It is for this reason that one of ordinary skill in the art would have been motivated to include acquiring the write lock on the row comprises acquiring a write lock that will prevent another statement from accessing the row while the statement is being processed.

34. Claims 24 and 35 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Nojima** (US 2004/0199540) in view of **Narang** (US 2003/0069902).

35. Regarding claim 24, **Nojima** teaches a system for executing a file system statement in the context of a transaction (See page 3, paragraph [0041] "The DBMS executes the process in a unit of transaction."), the file system statement including a call to open an item [DB managed contents], one of a call to read from the item [DB managed contents], and a call to write to the item [DB managed contents], and a call to close the item [DB managed contents] (See page 5, paragraph [0075] "A data read function is for acquiring data of a DB managed contents 243." And see page 6, paragraph [0082] "A data write function is for performing additional write or update of data for the DB managed contents 243.") the system comprising:

a relational data engine comprising a data table having a row corresponding to the item (See page 1, paragraph [0009] "By defining a table having fields of the file abstract data type, the DBMS user can register an external contents in the database as a link destination in the table.");

a storage platform built on the relational data engine, the storage platform comprising means for associating the file system statement with the transaction, (See page 5, paragraph [0070] "A transaction settling function is to settle link registration, link delete and content update for an external contents.").

Nojima does not explicitly disclose means for receiving the file system statement, and means for starting the transaction in response to receiving the file system statement by acquiring either a read lock or a write lock on the row.

However, **Narang** teaches means for receiving the file system statement (See page 2, paragraph [0020] "The DLFF 10 intercepts certain of the file system API calls 70 issued by the application. The file-system API calls that are intercepted include file-open, file-rename, and file-delete calls.") and means for starting the transaction in response to receiving the file system statement by acquiring one of a read lock and a write lock on the row. (See page 1, paragraph [0004] "...a transaction coordinator typically ensures a consistent view by locking out readers of meta-data as well as file data until the transaction is committed." This is referring to locking out database access during the transaction, which occurs by readers of the meta-data, which would have been initiated after intercepting the file system statement mentioned above.)

It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Nojima** with those of **Narang** because both address the field of database consistency related to external objects, and by including the read and write locks as disclosed in **Narang**, consistency control can be maintained while transactions are being processed. It is for this reason that one of ordinary skill in the art would have been motivated to include means for receiving the file system statement, and means for starting the transaction in response to receiving the file system statement by acquiring either a read lock or a write lock on the row.

36. Regarding claim 35, **Nojima** teaches a system substantially as claimed.

Nojima does not explicitly disclose the row comprises a filestream field.

However, **Narang** discloses the row comprises a filestream field. (See page 2, paragraph [0016] "The name column 83 typically contains a string, the department column an integer, while the picture column would contain a reference to an image stored in one of a number of external data files 20." The picture field in the reference is the same thing as the filestream field of the claim.)

It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Nojima** with those of **Narang** because both address the field of database consistency related to external objects, and by including a filestream field as disclosed in **Narang**, large file that normally would not be in the database can still be referenced from the database, providing for a more robust system. It is for this reason that one of ordinary skill in the art would have been motivated to include the row comprises a filestream field.

37. Claim 25 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Nojima** in view of **Narang** as applied to claim 24 above, and further in view of **Reed** (US 7,035,874).

Nojima and **Narang** teach a system substantially as claimed.

Nojima and **Narang** do not explicitly disclose the row corresponding to the item includes a user defined type corresponding to the item.

However, **Reed** teaches the row corresponding to the item includes a user defined type corresponding to the item. (See column 3, lines 48-51 "SQL also supports user defined types (UDT) and user defined methods (UDM). In one implementation, DBMS 100 supports SQL and includes support for UDT's and UDM's." and column 4, lines 41 – 43 "The cursor locks a selected portion of a table in DBMS 100 including the indicated media object (e.g., locking one or more rows).")

It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Nojima** and **Narang** with those of **Reed** because they all address the field of database consistency related to external objects, and by including the UDT's as disclosed in **Reed**, a more robust method is created because a variety of types of complex data can be referred to. It is for this reason that one of ordinary skill in the art would have been motivated to include the row corresponding to the item includes a user defined type corresponding to the item.

38. Claims 26 – 28 and 30 – 34 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Nojima** in view of **Narang** as applied to claim 24 above, and further in view of **Bamford** (US 5,870,758).

39. Regarding claim 26, **Nojima** and **Narang** teach a system substantially as claimed.

Nojima and **Narang** do not explicitly disclose associating a second statement [one or more] with the transaction.

However, **Bamford** teaches a means for associating a second statement with the transaction (See column 1, lines 20-21 "A logical unit of work that is comprised of one or more database language statements is referred to as a transaction.")

It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Nojima** and **Narang** with those of **Bamford** because they all comprise database methods for transaction concurrency and by including more than one statement within the transactions, it becomes more realistic as database transactions usually comprise more than one statement. It is for this reason that one of ordinary skill in the art would have been motivated to include a means for associating a second statement with the transaction.

40. Regarding claim 27, **Narang** additionally discloses the second statement is another file system statement. (See page 2, paragraph [0019] "The application 30 can now access the external data file 20 directly using standard file-system API calls 70. Typical file-system API calls 70 are 'file-open', 'file-read' and 'file-delete'.")

41. Regarding claim 28, **Nojima** additionally discloses the second statement is a transactional query language statement. (See page 4, paragraph [0062] "Described in (5) is a SELECT statement for acquisition of an access handle for accessing an external contents." Select is an example of a transactional query language statement.)

42. Regarding claim 30, **Nojima** and **Narang** disclose a system substantially as claimed.

Nojima and **Narang** do not explicitly disclose the read lock provides a read committed view of the row.

However, **Bamford** discloses the read lock provides a read committed view of the row. (See column 5, line 66 – column 6, line 7 "According to an embodiment of the invention, the SQL92 isolation levels READ COMMITTED and SERIALIZABLE are implemented using what shall be referred to hereinafter as 'Read Consistent Mode'. Read Consistent Mode is characterized by two rules. First, every statement executed by transaction sees only (1) changes that were committed to the database by a particular set of committed transactions (the 'snapshot set') and (2) changes made by the transaction itself.")

It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Nojima** and **Narang** with those of **Bamford** because they all comprise database methods for transaction concurrency and by including the read committed view as disclosed in Bamford, the database has an additional tool needed to retain consistency. It is for this reason that one of ordinary skill in the art would have been motivated to include the read lock provides a read committed view of the row.

43. Regarding claim 31, **Nojima** and **Narang** disclose a system substantially as claimed.

Nojima and **Narang** do not explicitly disclose the write lock that prevents another transaction from accessing the row while the transaction is being processed.

However, **Bamford** discloses the write lock that prevents another transaction from accessing the row while the transaction is being processed. (See column 6, lines 9 – 12 "Second, update transactions lock the rows they write, and hold those locks until the transactions complete. A lock on any given row may be held by only one transaction at a time.")

It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Nojima** and **Narang** with those of **Bamford** because they all comprise database methods for transaction concurrency and by including the write lock on the row as disclosed in **Bamford**, the database has an additional tool needed to retain data consistency. It is for this reason that one of ordinary skill in the art would have been motivated to include the write lock that prevents another transaction from accessing the row while the transaction is being processed.

44. Regarding claim 32, **Nojima** and **Narang** disclose a system substantially as claimed.

Nojima and **Narang** do not explicitly disclose the write lock prevents a non-transacted file system statement from accessing the row while the transaction is being processed.

However, **Bamford** discloses the write lock prevents a non-transacted file system statement from accessing the row while the transaction is being processed.

(See Column 3 lines 12 – 17 "Dirty write: only one transaction can hold a write lock on a specific data item. To preclude dirty writes, write locks must be held until a transaction commits. Virtually all database systems always prevent dirty writes, in order to support transaction rollback and prevent totally unpredictable and chaotic results.")

It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Nojima** and **Narang** with those of **Bamford** because they all comprise database methods for transaction concurrency and by including non-transacted file system statement restriction from row access as disclosed in Bamford, the database has an additional tool needed to retain data consistency. It is for this reason that one of ordinary skill in the art would have been motivated to include the write lock prevents a non-transacted file system statement from accessing the row while the transaction is being processed.

45. Regarding claim 33, **Nojima** and **Narang** disclose a system substantially as claimed.

Nojima and **Narang** fail to disclose the write lock prevents another statement within the transaction from writing to the row.

However, **Bamford** discloses the write lock prevents another statement within the transaction from writing to the row. (See column 6 lines 48 – 52 "For example, Read Consistent Mode with transaction-level snapshots may be implemented such that the snapshot set for all queries in a transaction includes all of the transactions that committed prior to the beginning time of the transaction." Because this is based on the

transaction level, between individual queries, other statements will not be able to be write until the entire transaction has finished.)

It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Nojima** and **Narang** with those of **Bamford** because they all comprise database methods for transaction concurrency and by including a write lock as disclosed in Bamford, the database has an additional tool needed to retain data consistency. It is for this reason that one of ordinary skill in the art would have been motivated to include the write lock prevents another statement within the transaction from writing to the row.

46. Regarding claim 34, **Nojima** and **Narang** disclose a system substantially as claimed.

Nojima and **Narang** fail to disclose the write lock enables another statement within the transaction to read from the row.

However **Bamford** discloses the write lock enables another statement within the transaction to read from the row. (See column 6, line 38-41 "For example, Read Consistent Mode with query-level snapshots may be implemented such that the snapshot set for each query includes all transactions that committed prior to the beginning time of the query." Because this is based on individual queries, rather than transactions, queries that occurred within a transactions will not be locked from being read.)

because they all comprise database methods for transaction concurrency and by including a write lock as disclosed in Bamford, the database has an additional tool needed to retain data consistency. It is for this reason that one of ordinary skill in the art would have been motivated to include the write lock enables another statement within the transaction to read from the row.

46. Claim 29 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Nojima** in view of **Narang** as applied to claim 24 above, and further in view of **Ponnekanti** (6,606,626).

Nojima and **Narang** teach a system substantially as claimed.

Nojima and **Narang** do not explicitly disclose determining whether starting the transaction will result in a conflict; if so, then resolving the conflict according to a conflict resolution scheme; and if not, then starting the transaction.

However, **Ponnekanti** teaches determining whether starting the transaction will result in a conflict (See column 4, lines 3 – 7 "The instant duration lock is a mechanism that allows the client requesting the lock to see whether there exists a conflicting lock already held on the row (i.e. from another concurrent transaction.); if so, then resolving the conflict according to a conflict resolution scheme (See column 4, lines 10-18 where a conflict resolution scheme is described); and if not, then starting the transaction [delete]. (See column 4, lines 7- "If no conflict is found, the 'lock instant' request will be granted and the client will know that the 'delete' has committed.")

It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Nojima** and **Narang** with that of **Ponnekanti** because they all comprise database methods for transaction concurrency and by including a conflict resolution scheme as described in **Ponnekanti**, the system becomes more robust and able to deal with errors. It is for this reason that one of ordinary skill in the art would have been motivated to include determining whether starting the transaction will result in a conflict; if so, then resolving the conflict according to a conflict resolution scheme; and if not, then starting the transaction.

Conclusion

47. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Dennis L. Vautrot whose telephone number is 571-272-2184. The examiner can normally be reached on Monday-Friday 9:00-6:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cottingham can be reached on 571-272-7079. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Dv
1 March 2007

JOHN COTTINGHAM
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100